The Study of Quantum Platform in Underwater Glider

Shengbo Qi, Zengchuan He, Ying Chai, Lele Yao, Chenglun Zhang, Dalei Song
College of Engineering, Ocean University of China, Qingdao 266100, China

***Abstract*—The software of traditional embedded systems have many shortcomings: the lack of openness, system maintenance difficulties, etc. This paper proposes a solution based on Quantum Platform to solve these problems, and applies it to glider system. The hierarchical state machines of main active object in underwater glider have been designed. Implementation process and test results show that this design ensures the reliability, stability and the quality of real-time of the system, improves the flexibility, maintainability and the quality of system software development.**

Keywords-underwater glider; embedded systems; Quantum Platform(QP); hierarchical state machine

I. INTRODUCTION

Underwater glider is a new ocean observation platforms, it relies on changing the position of gravity centre to change their attitude, changing buoyancy for motion. Each certain time interval, it surfaces for communication, receives instructions and starts the next operating cycle to dive and rise. Due to no power needed in motion, it can do a long, large-scale profile monitoring to the marine environment.

The electrical control system of Underwater Glider is a typical embedded system, which controls attitude, sampling, communications, and emergency treatment. Most of time, glider runs under water, and can't be contacted with control centre. Once bottomed or failed, it can't be located and recovered. So, the designer must consider instantaneity, reliability and stability of the system, which are determined by both hardware and software. With the development of microelectronics technology, embedded hardware becomes more powerful and standardized. The development of embedded software still has some problems, such as tight dependence between components, lacks of openness, software quality depends on the level of the programmers, etc [1-3]. Although operating systems and other ways can be used to improve software quality [4-6], yet the foreground/background system and the ordinary real-time operating system are all sequential control, which means program may actively suspend or passively blocked by signals [7]. The sequential control method is feasible in many areas, but not very satisfied in some cases, especially when there are several events which will arrive uncertain cycles and uncertain orders, but require real-time processing, just as the underwater glider electric control system.

A better solution is changing software architecture to event-driven model, as the QP. We established state machine model based on QP in this paper, which solved instantaneity, stability and reliability of the software system. Also it ensures code quality in this paper with the direct transition from model to code .

II. REQUIREMENTS AND ANALYSIS

A. Glider system components

The electrical system of glider includes four subsystems: for navigation, for motion, for information sampling, and for emergency.

Navigation subsystem is responsible for transferring data and instructions with shore station. Once glider surfaced, it passes current GPS location information to shore station and receives the instructions for next motion cycle.

Motion subsystem can control the buoyancy, the pitch and the roll of glider. This subsystem receives instructions from navigation subsystem and information sampling subsystem, and adjusts attitude and buoyancy.

Information sampling subsystem is mainly responsible for data sampling and storing, and the data includes depth, attitude and other sensor data. After sampling, it will pass a message to motion subsystem.

Emergency subsystem can do unload action, make glider surfaced, and send emergency signals to the shore station if there are something dangerous occurred.

B. Qp and uml requirements analysis

QP is a lightweight state machine engine, provides model-driven framework. The software system based on QP can be divided into three layers: board-level driver layer, QP layer and software application layer, and layers can be communicated by standard API function interface. Board-level driver layer includes drivers and API interface, which makes the communication between target board and application layer is transparent. The QP layer acts as bus, connecting board-level driver layer and software application layer. Software application layer is the implementation of system model.

The quality of system model directly affects system stability and reliability. UML is an object-oriented analysis method and a standardized modelling language. It can describe complete and accurate blueprint of a software system. UML includes a series of diagram， which can describe static constitutes and dynamic behaviours of a system [8]. For the UML is supported by QP, so the glider system is described and modelled by UML.

Use Case diagram is mainly used for functional description. Activity and Sequence diagrams are to describe the dynamic behaviours of system.

© 2015. The authors - Published by Atlantis Press

UML use case analysis

Figure 1 shows the overall Use Case of glider system. As the navigation controller, he need not know the specific steps of every motion, but only give the navigation instructions. All operations are done automatically by motion system. Navigation controller input navigation commands before system working.

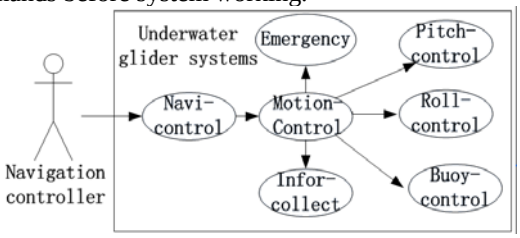


Figure 1. Use case diagram

Activity diagram describes general overview of system, as shown in Figure.2. We can separate various activities of sub-tasks from activity diagram clearly. Activity diagram provides stability and reliability assurance to the system.

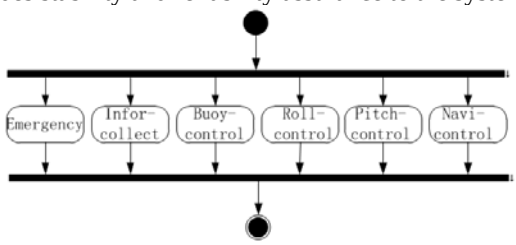


Figure 2. Activity diagrams

UML use case refinement

Only further refinement of cases and activity diagrams can provide clear guidance and specification. The buoy-control use case is an example.

Buoyancy is changed by the flow of oil between inner and outer capsule which caused by controlling pump and solenoid valve, which control the rising or diving of glider. A pressure sensor is embedded in the inner capsule.

Use Case name: Buoyancy

Use Case annotation: buoyancy adjustment

Initial conditions: system is running, with threshold value

Trigger Condition: the change of current oil pressure value or threshold value

Primary actor: pump

Secondary actor: solenoid valve

Main process:

Step1: Detecting the pressure of inner capsule;

Step2: Comparing inner pressure value with threshold value;

Step2.1: If inner pressure value is smaller than threshold value, then shut pump, open valve, and jump to step 1;

Step2.2: If inner pressure value is larger than threshold value, then shut valve, open pump, and jump to step1;

Step2.3: If inner pressure value is equal to threshold value, shut pump, close valve, and jump to step3;

Step3: Buoyancy adjustment completion.

UML sequence diagrams

Sequence diagrams drawing is a starting point to study event-driven system [9]. The drawing process is the process of ensuring stability and reliability of system software. In the process, it is necessary to break the coupling relationship between active objects. The coupling relationship is mainly related to public resources and the number of event interaction. The requirement is consistent with the characteristics of quantum framework.

Quantum framework (QF) is a component of QP and an application framework based on event-driven concurrent state machine. QF plays the role of "software bus" in system [8]. The only means of communication between active objects is QF, instead of share public resource. QF is equal to "middleware".

Four active objects are consistent with four subsystems of glider: navigation active object, motion active object, information sampling active object, and emergency active object. Among them, motion active object can be divided into three main states according function: buoyancy state, roll state, and pitch state. Each active object generates signals and sends to QF. And each active object accepts signals only from the QF. This feature ensures the stability and reliability of software.

Figure. 3 is Buoyancy regulator sequence diagram.

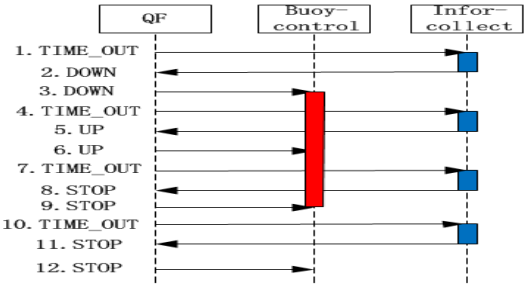


Figure 3. Sequence diagram

Regulate buoyancy process only involves QF, information sampling active object and buoyancy state of motion control active. QF is software bus, which is the target of signals published to and subscribed to. The red bar in the diagram is cycle of buoyancy regulation, and the blue bar is the cycle of information sampling.

The signals are as follows.

1. TIME_OUT: QF transmits a time event to information sampling object, and information sampling active object begins to collect information.

2. DOWN: Information sampling active object generates DOWN signal, and publishes it to quantum framework (QF).

3. DOWN: QF sends DOWN signal to motion active object, and motion active object begins to decrease buoyancy.

4.UP: Information sampling active object generates UP signal, and publishes it to quantum framework(QF).

6. UP: QF sends UP signal to motion active object, and motion active object begins to increase buoyancy.

8. STOP and 11. STOP: Information sampling active object generates STOP signal, and publishes it to quantum framework (QF).

9. STOP and 12.STOP: QF sends STOP signal to motion active object, and motion active object begins to stop to regulate buoyancy.

III. APPLICATION DESIGN

A. *Signal and active object*

Sequence diagrams can show the life cycle and motion law of active object. However, the choice of signal is the key of design, is the main factor to meet instantaneity. The more streamlined and fully of the signal classification, the higher efficiency of code and the stronger instantaneity that system can get.

There are two kinds of signals in quantum framework, one kind is for the exchange between active objects and the other is used to adjust internal state jump. For example, the signals generated by information sampling active object, guides buoyancy, roll and pitch regulation; while signals generated by time tick only regulate the state jump in a active object.

Here is part code of signals:

```
EnumGlider Signals{
                ROLL=Q_USER_SIG,
                B_R_SIG,
              MAX_PUB_SIG,
              TIME_OUT_SIG,
              MAX_SIG.
        };
```

The user-defined signals must begin as Q_USER_SIG, to avoid duplicate with QEP internal signals [10];

MAX_PUB_SIG indicates the end of signals that published by application[11].

MAX_SIG indicates the end of the application enumeration signals.

Follows are the part of active object events:

```
typedef struct InforEvtTag{
        QEvent super;
        uint8_t Save_type;
        uint8_t
    Save_buffer[250];
        } InforEvt;
```

Follows are active object function declarations:

```
Navigation_ctor((Navigation
*)AO_Navigatuon);
Sport_ctor((Sport*)AO_ Sport);
Infor _ctor((Infor *)AO_Infor);
Emergency_ctor((Emergency*)AO
_Emergency).
```

B. *State machine design*

The main task of design based on quantum framework is state machine design of each active object. QF is compatible finite state machine and hierarchical state machine. For underwater glider system, using FSM will lead to "state explosion" problem, while the HSM is a better solution to this problem [1].

Motion Active Object state machine design

Motion is mainly active object of system, its hierarchical state machine as shown in Fig.4. According glider working condition, Motion active object can be divided into two state machines: BEGIN state and gliding SPORT state. SPORT state can also be divided into three sub-states, and they are PITCH state, ROLL state and BUOY state. Sub-state can inherit super-state's behavior.

In PITCH state, servo motor is controlled, and the mass is moved back and forth to change the pitch.

In BUOY state, buoyancy is changed by the volume change of inter capsule by controlling pump and solenoid valve.

In ROLL state, another servo motor is controlled to drive roll masses moves around.

State machine transitions are interpreted as follows:

i> BEGIN is initial state or end state. System can accept navigation control instructions, and into SPORT state.

ii > SPORT is motion control super-state, which is divided into three sub-states.iii> PITCH state is responsible for controlling pitch angle. It is divided into three sub-states: PIT_KEP, PIT_FOR and PIT_BAC. PIT_KEP means to keep glider current pitch angle, PIT_FOR means to drive pitch mass forward, increases depression, and PIT_BAC means to drive pitch mass backward, increase elevation.

iv > ROLL state is responsible for the control of movement curve. It is divided into three sub-states: ROLL_KEP, ROLL_LEF and ROLL_RIG. ROLL_KEP means to keep orientation, ROLL_LEF means to drive glider left deflection; ROLL_RIG means to drive glider right deflection. Note that the glider operation curve can only be changed in glider moving.

v > BUOY state is responsible for controlling pitch angle. It is divided into three sub-states: BUOY_KEP, BUOY_DOW and BUOY_UP. BUOY_KEP means to maintain buoyancy; BUOY_DOW means to start solenoid valve, reducing the buoyancy; BUOY_UP means to start pump, increases buoyancy.

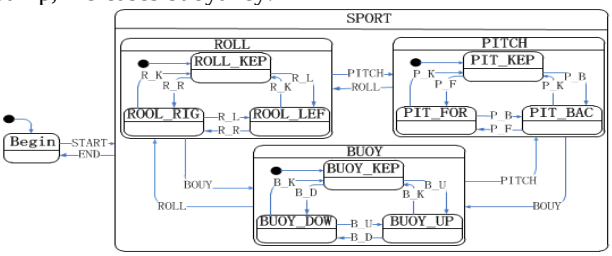


Figure 4. Hierarchical state machine of SPORT

We can have a conclusion, from the process of State machine design. The introduction of quantum platform weakened coupling-relations between active objects, and the whole state transition structure is clearer.

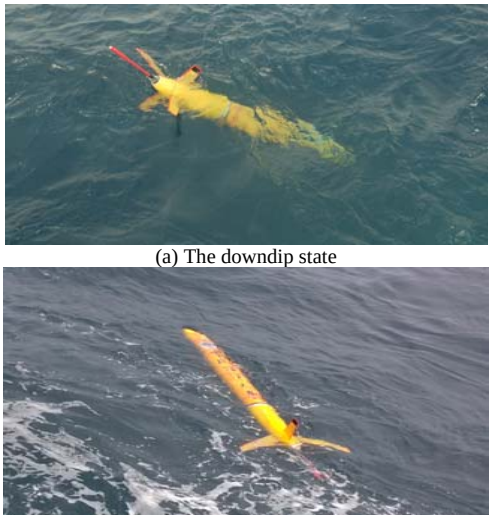
From UML model to QM model

QM (QP Modeler) is a free modeling tool, and can connect with Quantum platform seamlessly. With the automatic code generation feature, it can generate concise C code from state machine model directly.

UML state machines can be drawn in QM development environment, and transformed into C code directly, which makes the development cycle shorten. The QM tool enhances code execution efficiency, assures code quality and software reliability.

IV. IMPLEMENTION

In July 2014, the OUC-I glider was tested in Qingdao offshore. A series of experiments of glider were made to verify the effectiveness of hierarchical state machine based on QF, and all the necessary data were stored in the SD card.



(a) The downdip state

(b) The equilibrium State

Figure 5. Two states in moving (Qingdao offshore, China).

After that, data were read from SD card, and some of the glider motion processes were analyzed. Figure.6 is a profile analysis of the results of four sports glider. From the graph, we can have two relationships. The oil pressure becomes greater with the buoyancy smaller, and the glider's depth increases. The pitch decreasing means downward movement, and pitch increasing means upward movement.

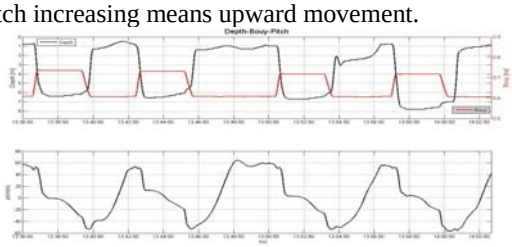


Figure 6. Multi-profile campaign data graph

From the figure, we know that the depth curve is W-shaped, glider's maximum dive depth is 8 meters, and glider's motion trajectory is basically same as the sets curve of shore station.

V. RESULTS

In this paper, we first analyzed the needs of glider, and then done modelling and analysis of use cases based on UML, designed the associated state machine. Finally, we completed the software design of underwater glider control system based on quantum platform. The experiments show that the software design is effective and successful, which meet the desired development goals.

ACKNOWLEDGEMENTS

The research work was supported by National High-Tech Research and Development Program (863 project) of China No. 2012AA091004.

REFERENCES

[1] QI Shengbo, Zhang Chengrui, Luo Ying, AMT Electronic Control System Based on SOPC and Quantum Framework. *Transactions of the Chinese Society for Agricultural Machinery*,2011,(11):13-19.

[2] LI Xiao-jun, ZHANG Cheng-rui. Research on Logic Modeling of Heavy Truck AMT Based on Quantum Framework(QF). *Journal of System Simulation*,2009,(13):3855-3859.

[3] XU Xiao-liang, WANG Le-yu, ZHOU Hong. Implementation framework of finite state machines. *Journal of Engineering Design*, 2003,(5):251-253.

[4] LiuSong, RESEARCH AND IMPLEMENTATION ON UNDERWATER GLIDER EMBEDDED CONTROL SYSTEM BASED ON QNX. Shenyang: Shenyang University of Technology,2007.

[5] WANG Shuxin, LI Xiaoping. Motion modeling and analysis of underwater glider, *Ocean Technology*,2005,(1):5-9.

[6] WANG Bingzhen, ZHU Guangwen. Study of Stab ility of Underwater Glider Non-acceleration Linearly Gliding,*Ocean Technology*,2009,(2):1-4.

[7] Miro Samek. *Practical Statecharts in C/C++ Quantum Programming for Embedded Systems*. Beijing: Beihang University Press, 2004.

[8] QI Shengbo, JI Fenglei,YU Jingdong. Modeling and Implementing of AMT Electronic Control System Based onUML, *Ocean Technology*,2013,(4):8-14.

[9] Dining Philosophers Problem (DPP) Example,2008.

[10] Reference Manual,Quantum Leaps, LLC,2008.

[11] Miro Samek. Practical UML Statecharts in C / C + + : event-driven programming for embedded systems. 2nd ed.Newnes.2008.